

**APPLICATION**  
**FOR**  
**UNITED STATES LETTERS PATENT**

**TITLE:            REPLICATED SERVICE ARCHITECTURE**

**APPLICANT:    Gary R. LAUTERBACH**

"EXPRESS MAIL" Mailing Label Number: EV436028338US

Date of Deposit: March 16, 2004

# REPLICATED SERVICE ARCHITECTURE

## Background

[0001] The functions of a computer are typically controlled by a central processing unit (“CPU”), commonly referred to as a processor. As processing demands increased, a single processor computer system was no longer considered sufficient. As a result, new computer system architectures evolved to include multiple processors housed within one computer system. Figure 1 shows such a prior art system architecture.

[0002] Typically, a multiprocessor system (100) includes a number of processors (*i.e.*, Processor A (102), Processor B (104), Processor C (106), and Processor D (108)) all connected by an interconnect (110). The interconnect (110) allows the processors (*i.e.*, Processor A (102), Processor B (104), Processor C (106), and Processor D (108)) to communicate with each other. Further, the interconnect (110) allows the processors to interface with a shared memory (112) and access other systems via a router (114).

[0003] A multiprocessor operating system (116) is typically used to control the processors (Processor A (102), Processor B (104), Processor C (106), and Processor D (108)). The multiprocessor operating system (116) provides a software platform upon which various services (118), for example, an e-mail server, a web server, a document management system, a database query engine, etc., may execute. More specifically, the multiprocessor operating system (116) receives requests from the various services and forwards the request onto the processors, which generate a response, and returns the response back to the requesting service, via the multiprocessor operating system (116).

[0004] Typically, the multiprocessor operating system (116) forwards requests to the processor designated as the master processor (in this example, processor A (102)). The processor designated as the master processor subsequently schedules the request to be processed on one of the other processors (*i.e.*, the slave processors (104, 106, 108)). After the scheduled slave processor has completed processing the request and generated a result, the result is returned to the master processor. The master processor subsequently returns the result, via the multiprocessor operating system, to the requesting service.

### Summary

[0005] In general, in one aspect, the invention relates to a system comprising a first node and a second node located in a single multiprocessor system, the first node comprising a first router and a first replicated service executing on a first operating system, the second node comprising a second router and a second replicated service executing on a second operating system, and a mesh interconnect connecting the first node to the second node using the first router and the second router.

[0006] In general, in one aspect, the invention relates to a system, comprising a first subset and a second subset located in a single multiprocessor system, the first subset comprising a first plurality of nodes and the second subset comprising a second plurality of nodes, wherein each of the first plurality of nodes and each of the second plurality of nodes comprises a router, and a replicated service executing on an operating system, a first mesh interconnect connecting the first subset to the second subset, a second mesh interconnect connecting each node in the first plurality of nodes to every other node in the first plurality of nodes, and a third mesh interconnect connecting each node in the second plurality of nodes to every other node in the second plurality of nodes.

[0007] Other aspects of the invention will be apparent from the following description and the appended claims.

### **Brief Description of Drawings**

[0008] Figure 1 shows a prior art system architecture.

[0009] Figure 2 shows a system in accordance with one embodiment of the invention.

[0010] Figure 3 shows a system in accordance with one embodiment of the invention.

[0011] Figure 4 shows a flow chart in accordance with one embodiment of the invention.

### **Detailed Description**

[0012] Specific embodiments of the invention will now be described in detail with reference to the accompanying figures. Like elements in the various figures are denoted by like reference numerals for consistency.

[0013] In the following detailed description of embodiments of the invention, numerous specific details are set forth in order to provide a more thorough understanding of the invention. However, it will be apparent to one of ordinary skill in the art that the invention may be practiced without these specific details. In other instances, well-known features have not been described in detail to avoid obscuring the invention.

[0014] In general, embodiments of the invention relate to a system having a replicated service architecture. More specifically, embodiments of the invention provide a multiprocessor system having one or more replicated services executing on two or more nodes. In one embodiment of the invention, the replicated services

are used to provide redundancy within the multiprocessor system such that when a first instance of a replicated service fails, the multiprocessor system still has access to a second instance of the replicated service. In this manner, embodiments of the invention enable a multiprocessor system having a replicated service architecture to continue to provide replicated services to a user even when one or more node in the multiprocessor system fail or become unavailable.

[0015] Figure 2 shows a system in accordance with one embodiment of the invention. More specifically, Figure 2 shows a multiprocessor system (100) having a node topology in accordance with one embodiment of the invention. In the embodiment shown in Figure 2, each node (*i.e.*, Node A (150), Node B (152), Node C (154), Node D (156), Node E (158)) in the node topology is connected to every other node. This node topology allows each node multiple communication pathways to communicate with any other node, such that if one or more nodes fail, the remaining nodes may still be able to communicate with one another. In one embodiment of the invention, a mesh interconnect (142) provides the communication hardware infrastructure (*i.e.*, the hardware to physically connect each of the multiple nodes in the node topology) for the node topology used within the system. Those skilled in the art will appreciate that other node topologies may be used that provide multiple communication pathways between each of the nodes in the node topology without requiring every node to be directly connected to every other node, as shown in Figure 2.

[0016] Continuing with the discussion of Figure 2, the individual components within each node are now described with respect to the exploded view of Node A (150). In one embodiment of the invention, each node includes a processor (*i.e.*, Processor A (160)), an associated memory (*i.e.*, Memory A (162)), an operating system (*i.e.*, operating system A (165) executing on the processor (*e.g.*, Processor A (160)), and one or more replicated services (*i.e.*, Replicated Service A (166)) executing on the node (*e.g.*, 150). Further, the node (*e.g.*, Node A (150)), in one

or more embodiments of the invention, interfaces with other nodes in the node topology using a router (*i.e.*, Router A (168)). In addition, in one embodiment of the invention, the node (*e.g.*, Node A (150)) may also include a cache (*i.e.*, Cache A (170)). The aforementioned components in the node (*e.g.*, Node A (150)) provide a means for each individual node to operate independently of the other nodes in the node topology.

[0017] In one embodiment of the invention, the hardware (*i.e.*, Processor A (160), Router A (168), Memory A (162), etc.) may be different for each node. For example, Processor A (160) in Node A (150) may be a Complex Instruction Set Computer (CISC) processor while the processor in Node E (158) may be a Reduced Instruction Set Computer (RISC) processor. Further, the operating system for each node may also be different. For example, Node A (150) may be running on a UNIX based operating system such as Solaris™ (Solaris is a trademark of Sun Microsystems, Inc.), while the operating system running on Node C (154) may be a Windows-based operating system such as Windows NT® (Windows NT is a registered trademark of the Microsoft Corporation.)

[0018] As described above, each node includes a set of replicated services (*e.g.*, Replicated Service Set A (166)). In one embodiment of the invention, the replicated services correspond to instances of services offered by the system (140). For example, the services may include, but are not limited to, e-mail servers, web servers, a document management system, a database query engine, etc. Thus, in one embodiment of the invention, a given service is said to be a replicated service if more than one instance of the service exists and is available on the system (140). In one embodiment of the invention, an instance of a service corresponds to a given application providing the service. Thus, a service is said to be a replicated service if two different applications executing on different nodes provide the service. For example, for the system (140) in Figure 2 to have a replicated web service, Node A (150) may run an Apache Web Server while Node B (152) may

run an Internet Information Server (IIS)<sup>TM</sup> (IIS is a trademark of the Microsoft Corporation). Those skilled in the art will appreciate that while Nodes A and B (150 and 152, respectively) were used in the above example, any pair of nodes within the system may host an instance of the replicated service.

[0019] Those skilled in the art will appreciate that the term “different application” does not require that the applications providing the replicated service be provided by separate companies or that they are necessarily different products. For example, the “different applications” may be different versions of the same application. Further, the “different applications” may be the same application but one instance is configured to run on a first operating system while a second instance is configured to run on a different operation system. As noted above, in one embodiment of the invention, the inclusion of replicated services allows multiprocessor systems to continue providing services to the multiprocessor system user(s) even when one or more nodes within the multiprocessor systems fails or becomes unavailable.

[0020] Continuing with the discussion of Figure 2, in one embodiment of the invention, the router (168) operates using a lightweight communication protocol that supports sending and receiving broadcast messages (or multicast messages) while not requiring large amounts of overhead (*e.g.*, large headers, etc). Alternatively, the router (168) may use a heavy-weight protocol such Transmission Control Protocol (TCP) and Internet Protocol (IP). Those skilled in the art will appreciate that depending on the node topology, the router (168) may also include an appropriate routing algorithm to allow for communication between the nodes. In addition, the router (168) may include functionality to forward data from one node to another node (*e.g.*, router (168) may include functionality to “pass-through” data received from Node E (158) to Node B (152)). Further, in one embodiment of the invention, the routing protocol is designed to operating

without requiring a master node to control the routing within the system, *i.e.*, the router implements a master-less routing policy.

[0021] Those skilled in the art will appreciate that bandwidth requirements to allow broadcast messages (or multicast messages) between nodes may vary depending on the choice of node topology and communication protocol. Accordingly, the mesh interconnect (142), and, more specifically, the bandwidth built into the mesh interconnect (142) may vary depending on the aforementioned factors.

[0022] As noted above, each node in the system may include a cache (*e.g.*, cache A (170)). In one or more embodiments of the invention, the cache associated with a given node may also include a data structure to provide information about the replicated services provided by the particular node. For example, the data structure may correspond to a table that includes an entry for each replicated service provided by the node. Though not shown in Figure 2, each node may also include an external I/O port to allow communication with processes and/or devices that are external to the system.

[0023] As shown in Figure 2, the system may include five interconnected nodes. This system, shown in Figure 2, may be used as a building block for a distributed system in which the system shown in Figure 2 is one of many subsets of node topologies that make up the larger system. Such a system is shown in Figure 3.

[0024] Figure 3 shows a system architecture in accordance with another embodiment of the invention. The system in Figure 3 includes a series of interconnected subsets (*i.e.*, subset A (181), subset B (180), subset C (182), subset D (184), subset E (186)) each connected by a mesh interconnect (185). Each of the subsets (181, 180, 182, 184, 186) may be implemented using the same node topology as described in Figure 2. Alternatively, each subset may have a different node topology. Those skilled in the art will appreciate that while the subset



topology shown in Figure 3 includes a direct connection between each pair of subsets, the invention may be implemented such that each subset has at least two communication pathways (direct or in-direct) to every other subset. Further, those skilled in the art will appreciate that the routing algorithms used by the routers within the individual nodes include functionality to traverse the mesh interconnect of the subsets and functionality to further traverse the other individuals nodes within the subsets.

**[0025]** Figure 4 shows a flow chart in accordance with one embodiment of the invention. Initially, a node requests a replicated service (Step 100). In one embodiment of the invention, the node requests a replicated service because the particular replicated service on the node requesting the replicated service has failed, is busy, or is unavailable for another reason. Additionally, the node may request the replicated service from another node(s) because the node requesting the replicated service does not currently provide the replicated service.

**[0026]** Continuing with the discussion of Figure 4, the node requesting the replicated service subsequently generates a request for a replicated service (Step 102). Depending on the communications protocol implemented in the multiprocessor system for the node, the request may be a broadcast request (or a multicast request), etc. After the request is generated, the request is subsequently sent to a first subset of nodes (Step 104). In one embodiment of the invention, the first subset of nodes may correspond to the nodes directly connected to the node requiring one or more replicated services. Alternatively, the first subset of nodes may include a set of nodes explicitly specified in the request, regardless of the location within the system. Alternatively, those skilled in the art will appreciate that the first subset of nodes may corresponds to any subset of nodes in the multiprocessor system.

[0027] Continuing with the discussion of Figure 4, after the request is sent, the node sending the broadcast message (or a multicast message) subsequently waits to receive a response from each node in the first subset of nodes. The response should indicate whether any one of the nodes in the first subset of nodes has the requested replicated service available (Step 106). In one embodiment of the invention, when a node within the first subset of nodes receives a request for a replicated service from another node, the cache associated with the node receiving the request is examined. If the replicated service is listed in the cache, then a response is sent to the node that sent the request. The response indicates the availability of the replicated service. Those skilled in the art will appreciate that, in some instances, if the replicated service is not listed in the associated cache, the node receiving the request (via the operating system), may query the replicated services on the that node received the request prior to responding to the request.

[0028] Alternatively, if the node receiving the request does not include an associated cache, then when a request for a replicated service is received, the node queries the replicated services currently executing on the node and determines whether any of the replicated services executing on the node correspond to the replicated services being requested. If a replicated service corresponding to the requested replicated service is present, then the node generates and sends a response to the node that requested the replicated service. The response indicates the presence/availability of the replicated service.

[0029] Continuing with the discussion of Figure 4, if the requested replicated service is found executing on any one of the first subset of nodes, then all subsequent requests for the replicated service are re-routed to the node that includes the replicated service (Step 108). Thus, referring to Figure 2, if the Node A (150) requested a particular replicated service and, via the aforementioned method, determines that Node B (152) includes that replicated service, then all

subsequent requests to Node A (150) for the particular replicated service are re-routed to Node B (152).

[0030] However, if the requested replicated service is not present in the first subset of nodes, then a subsequent request (similar to one described above in Step 102) is generated and broadcast (via a broadcast or a multicast message) to a wider set of nodes (Step 110). The node that sent the broadcast message (or multicast message) subsequently waits to receive a response indicating whether or not any one of the wider set of nodes includes the requested replicated service, as described with respect to Step 106 (Step 112). If the requested replicated service is found executing on the any one of the wider subset of nodes, then the node proceeds to performs the actions described above with respect to Step 108.

[0031] Alternatively, if the replicated service is not found, then the node requesting the replicated service determines whether any remaining nodes exist requiring a query (step 114). If there are additional nodes to query, then the node requiring the replicated service proceeds to perform the actions associated with Step 110. Alternatively, if there are no remaining nodes to query, the node halts sending any other messages. At this stage, if the replicated service is not available on any of the nodes, then node requesting the replicated service may wait for a period of time and repeat steps 100-114. Alternatively, the request to obtain replicated services may fail.

[0032] Those skilled in the art will appreciate that a give node may also not respond to a request for a replicated service if the node is heavily loaded or overload. Accordingly, embodiments of the invention may also be applied to load balancing in multiprocessor system.

[0033] As mentioned above, in one or more embodiments of the invention, the routers within the individual nodes include functionality to re-route network traffic from one node to another. Further, each node includes functionality to determine

the status any node in the system and to re-route the network traffic of any node in the systems. Thus, if a given node fails, the remaining nodes in the system are able to ascertain this fact and re-route network traffic to the remaining nodes, accordingly. Those skilled in the art will appreciate that the aforementioned functionality does not require a master processor. Rather, each node co-operates with the other nodes such that all the network traffic is re-routed to the appropriate nodes.

[0034] In one embodiment of the invention, all nodes within the multiprocessor system are governed by a set of rules that dictate how traffic is to be re-routed when a given node fails. These may be built into the nodes via software and/or hardware. Thus, when a given node fails, the remaining nodes, using the set of rules, is able to successfully re-route the network traffic without requiring a master node/processor.

[0035] The following example is included to illustrate potential uses of the invention. The examples are not intended to limit the scope of the application or the potential uses of the invention. In one embodiment, the invention involves a means to continue providing services when certain operating systems and instances of replicated services are unavailable. For example, consider the scenario in which a particular computer virus is designed to exploit a security flaw in a particular operating system; however, other operating systems without this flaw are unaffected.

[0036] In this scenario, a multiprocessor system (without replicated services and different operating systems) would be vulnerable to such a computer virus if the only operating system running on the multiprocessor system was targeted by the virus. However, the presence of different operating systems, replicated services, and isolated nodes provide a counter measure to offset the vulnerability of operating systems and replicated services to such a virus. Thus, even if one of the

nodes in the multiprocessor system, designed in accordance with the present invention, is vulnerable and fails in response to the virus, the services provided by the failed node may continue to be available at a different node running a different operating system, which is not vulnerable to the virus.

[0037] Thus, if one of the nodes fails, the processing being performed on the failed node may be re-routed to an unaffected node. In this manner, the present invention provides a robust redundant system to provide services even when one or more nodes fail. Similar benefits may also be seen in the area of Internet security where hackers may exploit certain security holes present in a given application or operating system. Similar to the virus scenario, the nodes that are executing operating systems or replicated services, which are resulting in a security breach may be “turned off” and the workload that the affected nodes were performing may be off loaded to the unaffected nodes.

[0038] While the invention has been described with respect to a limited number of embodiments, those skilled in the art, having benefit of this disclosure, will appreciate that other embodiments can be devised which do not depart from the scope of the invention as disclosed herein. Accordingly, the scope of the invention should be limited only by the attached claims.